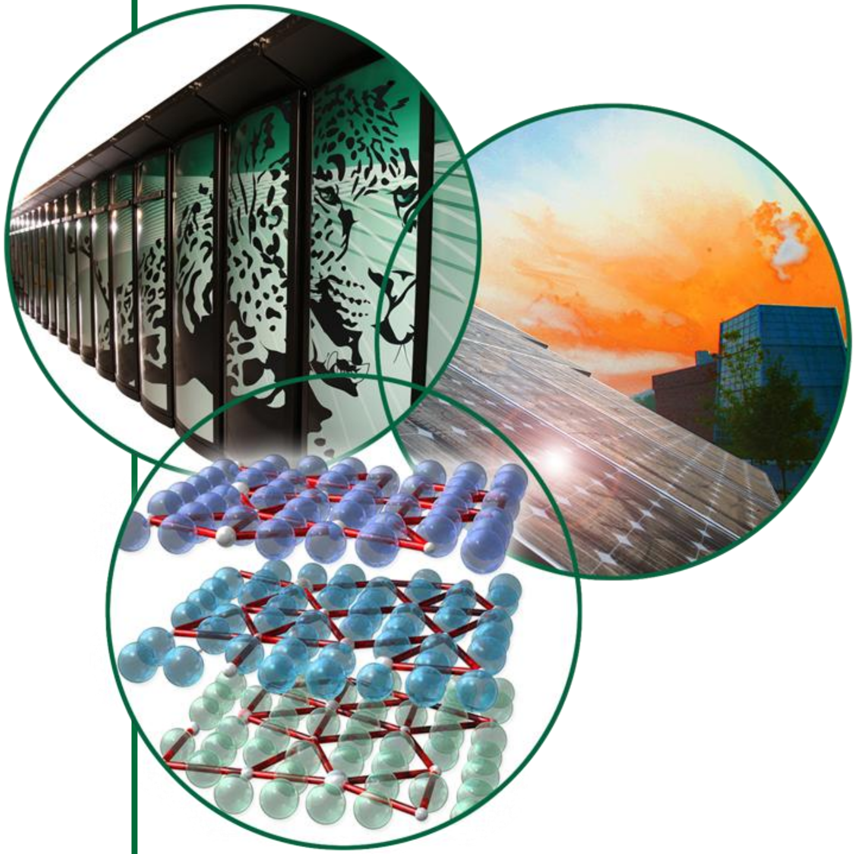
Vampir at ORNL

Trace-based Performance Analysis

Thomas Ilsche, ORNL

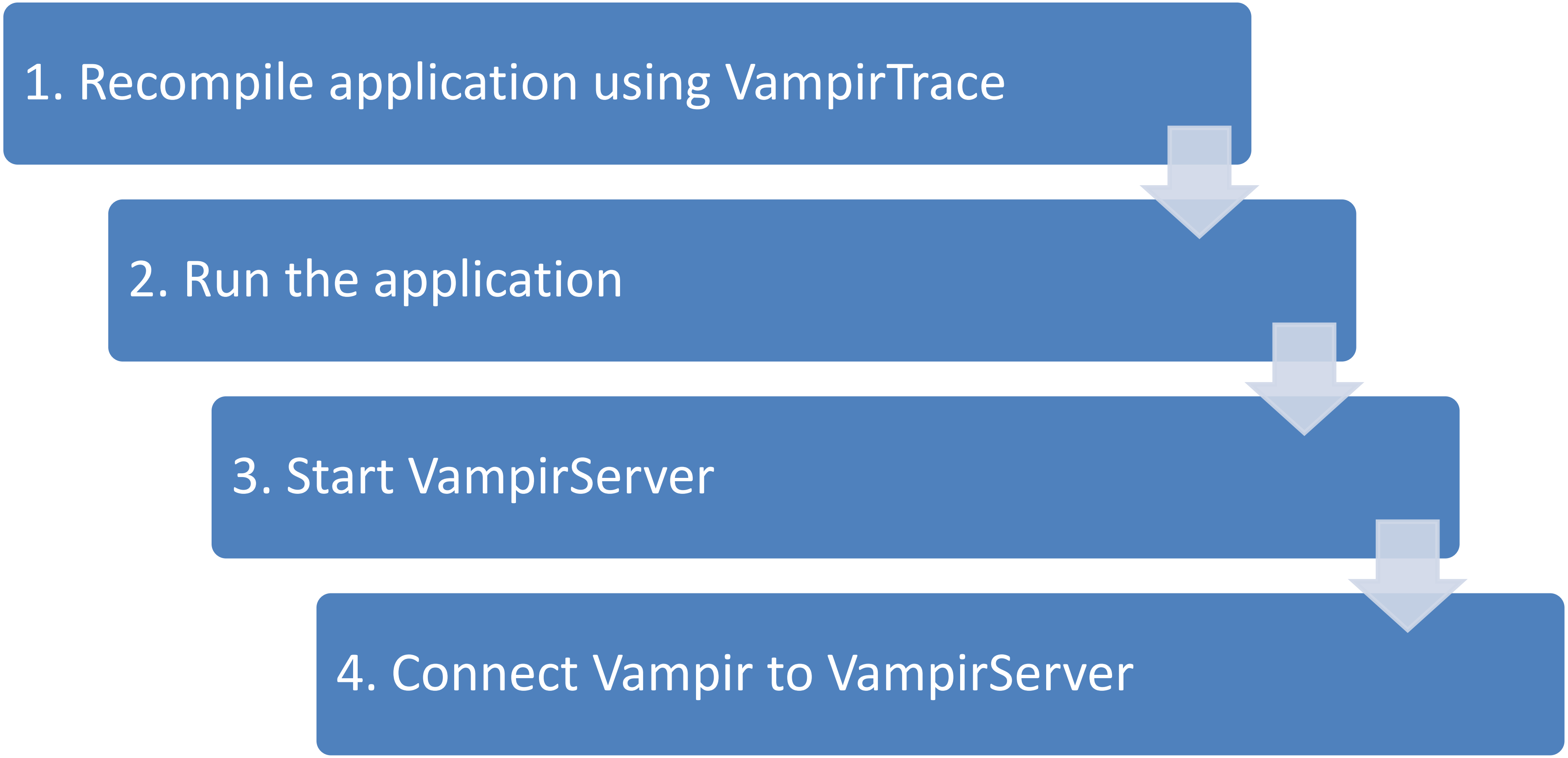
May 17, 2011



U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Fours steps to get you going



1. Recompile application using VampirTrace

2. Run the application

3. Start VampirServer

4. Connect Vampir to VampirServer

Step 1. Recompile application using VampirTrace

- First: `module load vampirtrace`
- Use the appropriate compiler wrappers
  - vtcc, vtCC, vtf77, vtf90
  - Pick appropriate library (seq, mpi, mt, hyb)
    - e.g. `vtcc -vt:hyb` (recommended)
  - Pick instrumentation type
    - `-vt:inst compinst` (default, compiler instrumentation, all functions)
    - `-vt:inst manual` (MPI, OpenMP, CUDA and manual)
  - See what's going on behind `-vt:show` or `-vt:verbose`
  - More details `-vt:help`
- Tell your build system to use VampirTrace
  - `./configure -with-CC=“vtcc -vt:hyb”` …

Step 2. Running the application

- Run your application as usual
- Make sure the VampirTrace module is loaded
  - Load module before and `qsub -V`
  - `module load vampirtrace` in the runscript
- Control VampirTrace by using environment variables
  - `VT_PFORM_GDIR=traces`
    - `mkdir $VT_PFORM_GDIR`
    - `lfs setstripe -c 1 $VT_PFORM_GDIR`
    - place trace files in that directory
    - Use /lustre/widow1 for large runs

Step 2. Running the application (contd.)

- Control VampirTrace by using environment variables
  - `VT_FILTER_SPEC=default.filter`
    - `cp $VAMPIRTRACE_DIR/etc/default.filter .`
    - Reduces resulting trace size by filtering frequently called functions
  - `VT_MAX_FLUSHES=20`
    - Defaults to 1
    - Allows VT to flush buffer during execution
  - `VT_BUFFER_SIZE=100M`
    - Defaults to 32M
    - Increases internal buffer size
  - `VT_SYNC_FLUSH=yes`
    - Use that if you have collective ops in your code
    - Avoids all processes waiting for one process to do trace I/O
    - But has overhead itsself (extra allreduce on all collective ops!)
  - `VT_METRICS=PAPI_FP_OPS:PAPI_TOT_INS`

Step 3+4. Analyze small traces locally (< 100 MB)

- Copy the Vampir GUI / Client to your workstation/laptop

  -
    ```
    scp home.ccs.ornl.gov:/sw/sources/vampir/client/vampir-7.*-i386.tar.gz .
    ```

  -
    ```
    tar –xzf vampir-7.*-i386.tar.gz
    ```

  -
    ```
    ./vampir/bin/vampir
    ```
- Client is available for:
  - Linux {i386, amd64}.tar.gz
  - Windows {x86, x64}.exe
  - MacOS as .dmg
- Copy trace files to your workstation/laptop

  -
    ```
    scp -r jaguarpf.ccs.ornl.gov:/tmp/…/traces/ .
    ```
- Start Vampir on your workstation/laptop
  - Open trace and enjoy!

Step 3. Start VampirServer

- VampirServer runs on the compute nodes and needs access to the generated trace files (.otf, .z)
- Currently available on jaguar, smoky, lens

-
  ```
  $ module load vampir
  $ qsub -V -I -l size=120,walltime=2:0:0 -A <ACC>
  $ vngd-start.sh
  ```
- Number of processes depends on (size of) the trace
  - Try ¼ of the cores used by the application
  - Running out of memory → add nodes
  - Getting strange MPI errors → reduce process count

Step 4. Connect Vampir to VampirServer (Linux/Mac)

- Open an SSH tunnel to the compute node

  -
    ```
    jaguarpf$ vngd-start.sh
    Launching VampirServer Version 2.3.0 with  on 12 processes ...
    Found license file: /tmp/work/tilsche/.vampir/etc/lic.dat
    Running 11 analysis processes... (abort with Ctrl-C or vngd-
    shutdown)
    Server listens on: nid13928:30051
    ```

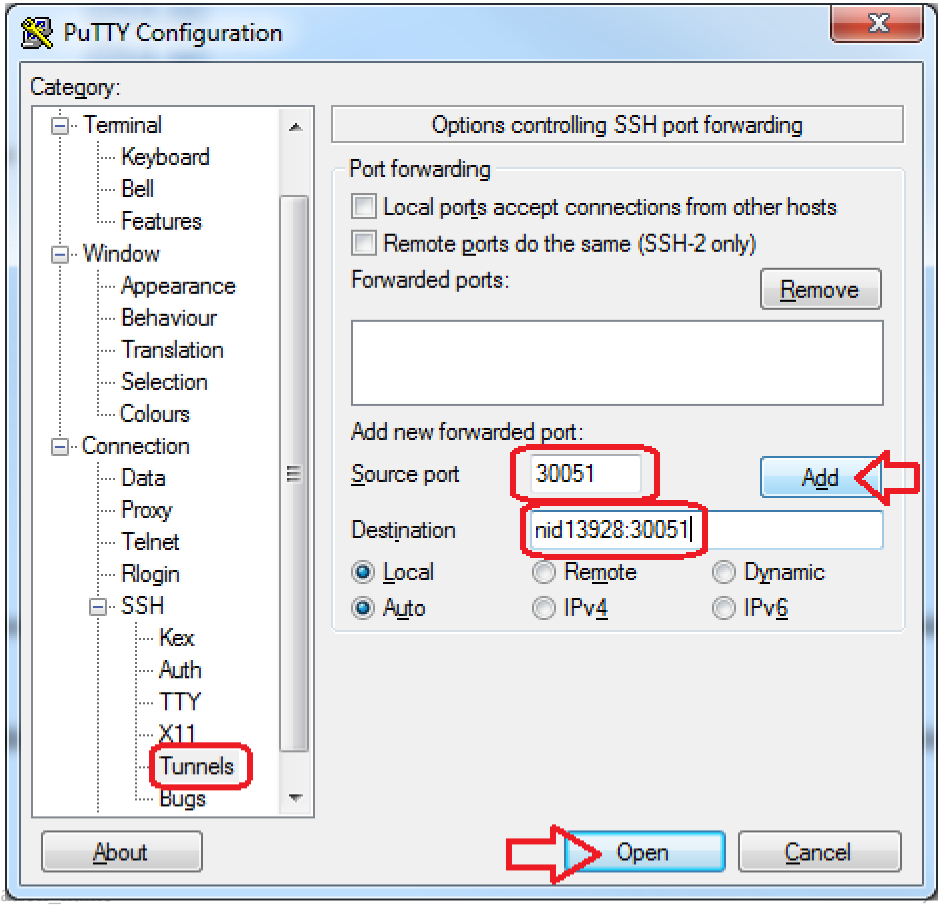
  -
    ```
    workstation$ ssh –L 30051:nid13928:30051 jaguarpf.ccs.ornl.gov
    ```
- Open Vampir and “remote open” to
  - Server: `localhost`
  - Port: `30051`

Step 4. Connect Vampir to VampirServer (Windows)

- We need an SSH tunnel to the compute node

  -
    ```
    jaguarpf$ vngd-start.sh
    Launching VampirServer Version 2.3.0 with  on 12 processes ...
    Found license file: /tmp/work/tilsche/.vampir/etc/lic.dat
    Running 11 analysis processes... (abort with Ctrl-C or vngd-shutdown)
    Server listens on: nid13928:30051
    ```

- Vampir “remote open” to
  - Server: `localhost`
  - Port: 30051



□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Source port: 30051

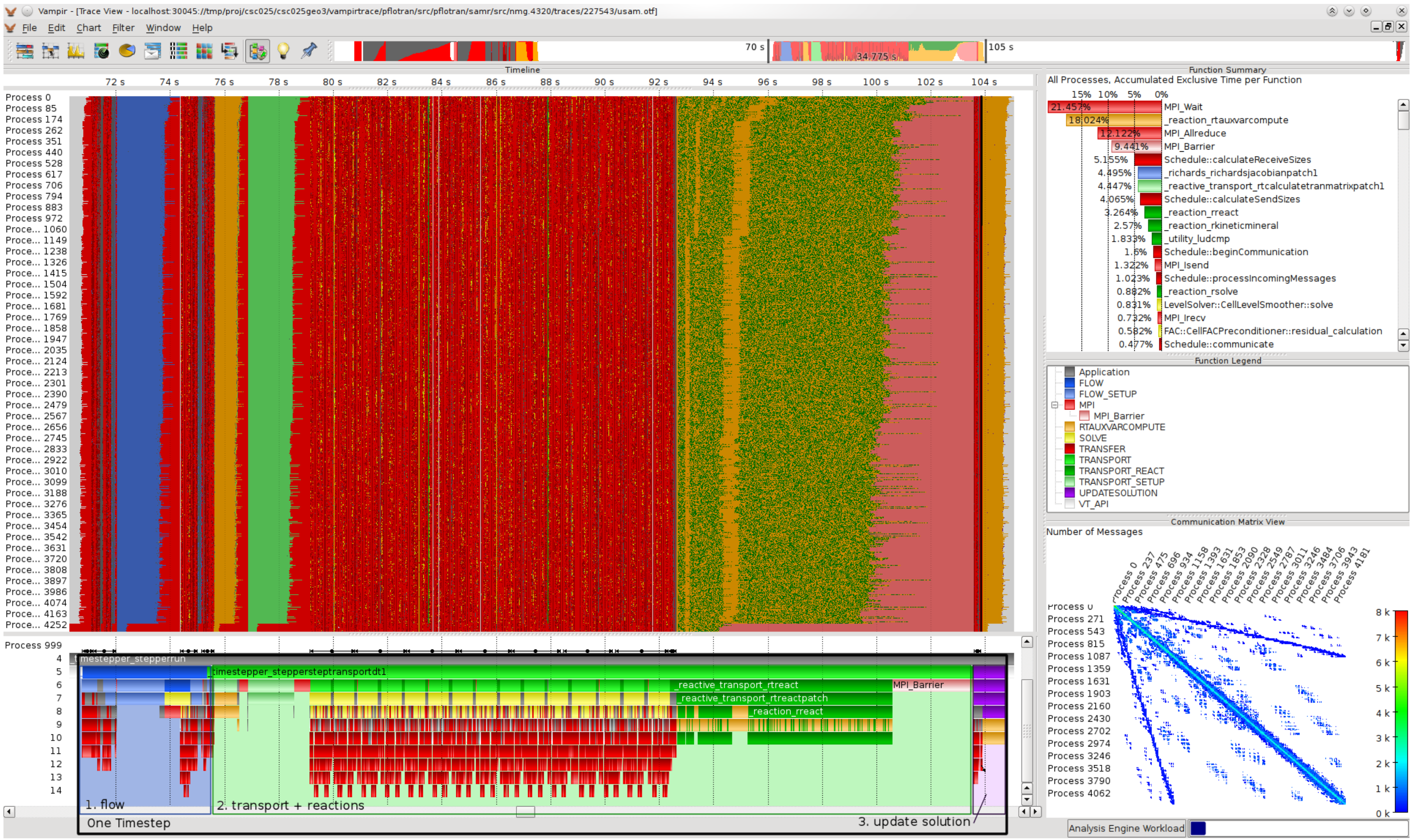
Destination: nid13928:30051

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

View your application trace:

- "Play around" with Vampir to get a feeling about the features



All Processes, Accumulated Exclusive Time per Function

21.457%	MPI_Wait
18.024%	_reaction_rtauxvarcompute
12.122%	MPI_Allreduce
9.441%	MPI_Barrier
5.155%	Schedule::calculateReceiveSizes
4.495%	_richards_richardsjacobianpatch1
4.447%	_reactive_transport_rtcalculatetranmatrixpatch1
4.065%	Schedule::calculateSendSizes
3.264%	_reaction_rreact
2.57%	_reaction_rkineticmineral
1.833%	_utility_ludcmp
1.6%	Schedule::beginCommunication
1.322%	MPI_Isend
1.023%	Schedule::processIncomingMessages
0.882%	_reaction_rsolve
0.831%	LevelSolver::CellLevelSmoother::solve
0.732%	MPI_Irecv
0.582%	FAC::CellFACPreconditioner::residual_calculation
0.477%	Schedule::communicate

Further information about VampirTrace & Vampir

- www.nccs.gov/computing-resources/jaguar/software/?software=vampir
- vampir-7.4.0-OLCF3/doc/Manual.pdf
- www.vampir.eu
- /sw/sources/vampirtrace/5.11ornl/doc/UserManual.pdf
- www.tu-dresden.de/zih/vampirtrace

Thank you

- Contact:
  - Thomas Ilsche, ORNL
    5700 B206
    tt1@ornl.gov
    865-241-6293

Live Demo:

- Show features of Vampir using a real application trace
- Application: Pflotran (AMR Physics) running on 4320 cores
- Total trace size 242 GB
- Using ~540 analysis processes on jaguar
- Different Instrumentation means
  - Compiler instrumentation
  - Exclude function lists for compiler instrumentation
  - Manual instrumentation
  - Manual control (disable tracing except for one particular timestep)
- Uses function groups and a color scheme